CoCoALib - Slug #777

SLUG: elimination

15 Sep 2015 22:31 - John Abbott

Status:	In Progress	**Start date:**	15 Sep 2015
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	10%
Category:	Improving	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	1.95 hour

Description

The following elim is very slow... why?

```
use QQ[x,y];
d := 12;
a:=1000;
f1:=x^d-(a*x-1)^2;
f2 := x^(d/2)*y^(d/2) - (y-x^(d/4))^2;
J := elim([x],ideal(f1,f2));  ---> takes way too long... why?

f := gens(J)[1];
RR := RealRoots(f);
```

Related issues:

Related to CoCoALib - Design #1326: Modify function myElim so that it returns...	**Closed**	**03 Oct 2019**
Related to CoCoA-5 - Slug #1270: RationalSolve: use MinPolyQuot instead of elim	**Closed**	**05 Apr 2019**
Related to CoCoALib - Slug #1394: Oddly slow GBasis computation (slow final c...	**Resolved**	**15 Jan 2020**
Related to CoCoALib - Slug #1796: myFinalizeGBasis ("Final clean up") should ...	**New**	**18 Mar 2024**

History

#1 - 15 Sep 2015 22:34 - John Abbott

I'm not sure how long the elim took (perhaps 20-30mins?)
The input is simple, the output not even that large (deg=72); why is it so slow?

The same elim modulo p is tolerably fast (perhaps 0.1 sec per prime?), and only a few primes would be needed to get the answer via CRT...

#2 - 16 Feb 2017 13:14 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I happened upon the file for the example, and tried it with SetVerbosityLevel(1999).
It seems that much of the time is spent in myFinalizeGBasis.

It would be nice to make elimination faster.

#3 - 16 Feb 2017 13:17 - John Abbott

The same example computed using MinPolyQuot is much faster: about 0.2s :-)

Maybe elim can be programmed to use MinPolyQuot when possible?

#4 - 17 Feb 2017 08:00 - Anna Maria Bigatti

Apart from the improvements we can do for elim (and are about to do ;-)
I think myFinalizeGBasis can be improved: this step is done to compute the reduced GBasis (in case of non-homogeneous input).

Now it reduces each element in myGB against myTrueReductors.
I have the feeling that there are elements in myGB whose LT is reducible, then those are surely to be removed (without wasting time in reducing them).
Extreme case: is 1 is in GB then myFinalizeGBasis should be instant.
I'll investigate.

#5 - 03 Oct 2019 17:19 - Anna Maria Bigatti

- Related to Design #1326: Modify function myElim so that it returns ideal? (not quite) added

#6 - 03 Oct 2019 17:24 - Anna Maria Bigatti

- Related to Slug #1270: RationalSolve: use MinPolyQuot instead of elim added

#7 - 03 Oct 2019 17:25 - Anna Maria Bigatti

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99800

#8 - 12 Mar 2021 09:57 - John Abbott

Has there been any progress on this issue?

In the case of elimination perhaps the "final clean up" could be limited to just those elems in the GB whose LTs contain only "keep" indets?
There would seem to be no point in "cleaning" GB elems which are anyway going to be discarded.

Could this be possible?

#9 - 03 Nov 2021 16:56 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#10 - 16 Feb 2024 18:03 - John Abbott

- Related to Slug #1394: Oddly slow GBasis computation (slow final cleanup) added

#11 - 18 Mar 2024 14:16 - Anna Maria Bigatti

- Related to Slug #1796: myFinalizeGBasis ("Final clean up") should be more flexible added

#12 - 22 Mar 2024 15:58 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

Anna Maria Bigatti wrote:

> Apart from the improvements we can do for elim (and are about to do ;-)

I wonder if we made them...

> I think myFinalizeGBasis can be improved: this step is done to compute the reduced GBasis (in case of non-homogeneous input).

> Now it reduces each element in myGB against myTrueReductors.
> I have the feeling that there are elements in myGB whose LT is reducible, then those are surely to be removed (without wasting time in reducing them).

Now I did it (a bit of chasing around and implementation of FindReducer(GPoly, TrueRed)). All tests passing.

> Extreme case: is 1 is in GB then myFinalizeGBasis should be instant.
> I'll investigate.

This has been done
(in myUpdateBasisAndPairs(const GPoly& inPoly) Revision 1.99 2018/03/08)

#13 - 22 Mar 2024 17:25 - Anna Maria Bigatti

Now it takes 165s (I don't know how long it was before)

#14 - 22 Mar 2024 18:12 - Anna Maria Bigatti

There are two improvements we can still make:

1. tell myFinalizeGBasis interreduce only the polynomials with the wanted indets (#1796)
2. use MinPoly when possible (#777-3)

#15 - 22 Mar 2024 18:12 - Anna Maria Bigatti

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880